*Proceedings of the 10th Mediterranean Conference*
*on Control and Automation - MED2002*
*Lisbon, Portugal, July 9-12, 2002.*

# Application of Fuzzy-Linguistic Geometry on Robot 3D Path Planning

**S. Khanmohammadi[*], A. Aghagolzadeh[**], M. A. Badamchi Zadeh[†], S. Ghaemi[††]**

Email: [*] khan@tabrizu.ac.ir [**] aghagol@tabrizu.ac.ir
† mbadamchi@tabrizu.ac.ir †† sgaemi@tabrizu.ac.ir

**Keywords:** Linguistic Geometry ; 3D Path Planning ; fuzzy

## Abstract

Linguistic Geometry is a new method in artificial intelligence which it's utilization causes a considerable reduction in the number of branches in a search tree. It has proved efficient in solving a variety of search problems such as path planning for a moving robot.

One of the short comings of Linguistic Geometry appears when the target is surrounded by static obstacles, consequently the robot can't detect any path to reach it's target and stays in initial point. In this paper a fuzzy procedure is used to overcome this shortage.

## 1. Introduction

The classic approach based on the theory of differential games is insufficient for large scale systems such as multirobot path planning, especially in case of dynamic. This method is expand to a discrete-event mode to be implemented as a purely interrogative simulation (Rodin, 1988; shinar, 1990). This discrediting leads to a finite game tree. The nodes of the tree represent the states of the game, where the players can select their controls for a given period of time. It is also possible that players do not make their decisions simultaneously and in this case, the sequential mutual movements can be easily distinguished. Thus, the branches of the tree are the moves in the game space. The pruning of such tree is the basic task of heuristic search techniques. Interrogative approach to control problems offers much faster execution and clearer simulator definition (Lirov et al., 1998). For this kind of approach a series of hierarchical dynamic multiagent goal-oriented systems should be developed and investigated.

One of the basic ideas is to decrease the dimension of the real-word system following the approach of a human expert in the field, by breaking the system into smaller subsystems. This process of decomposition can be applied recursively until we end up with a collection of basic subproblems that can be treated (in some sense) independently.

In the beginning of 80's Botvinik, Stilman, and others developed one of the most interesting and powerful heuristic hierarchical models. It was successfully applied to scheduling, planning, control, and computer chess. The hierarchical networks were introduced in (Botvinnik, 1984) in the form of ideas, plausible discussions, and program implementations. This model is considered as an ideal case for transferring the developed search heuristics to other domains employing formal linguistic tools.

An application of the developed model to a chess domain was implemented in full as program PIONEER (Botvinnik, 1984). To discover the inner properties of human expert heuristic, which is successful in a certain class of complex control systems, Stilman has developed a formal theory, the so called Linguistic Geometry.

By using this method, there will always be a considerable reduction in the number of branches of the search tree, this reduction is more observable in three dimensional pass planning problems [6,7].

For example in typical problems number of searches decreases from one billion branches to 56 only. Before introducing of fuzzy LG method a brief review of LG presenting in the following sections [1].

## 2. Complex System

Definition: A Complex System is presented by the following eight-tuple [1]:

$$< X , P , R_p , \{ON\} , v , S_i , S_t , TR > \quad (1)$$

where:

$X = \{x_i\}$ is a finite set of points (locations of elements);

$P = \{p_i\}$ is a finite set of elements (a union of two nonintersecting subsets $P_1$ and $P_2$);

$R_p(x, y)$ is a set of binary relations of reachable in $X$ ($x$ and $y \in X$ and $p \in P$) ; Element $p$ can move from point $x$ to point $y$ if point $y$ is reachable from $x$ , i.e. $R_p(x, y)$ holds.

$ON(p) = x$, where $ON$ is a partial function of placement from $P$ into $X$ ;

$v$ is a function on $P$ with positive integer values describing the values of elements;

$S_i$ , $S_t$ are the descriptions of the initial and target states. The Complex System searches the state space, which should have initial and target states;

$TR$ is a set of operators , $TRANSITION(p, x, y)$ of transitions of the system from one state to another.

The problem of the optimal operation of the system is considered as a search for the optimal sequence of transitions leading from the initial state $S_i$ to a target state $S_t$ based on certain criteria for optimality.

## 3. Autonomous Robots as Elements of the Complex System

A robot control model can be represented as a Complex System.

$X$ represents the operational district [3];

$P$ is the set of robots or autonomous vehicles;

$R_p(x, y)$ represent moving capabilities of different robots;

$ON(p) = x$ if robot $p$ is at point $x$;

$v(p)$ is the weighting value of robot $p$;

$S_i$ is a set of arbitrary initial or starting states;

$S_t$ is the set of target states;

$TRANSITION(p, x, y)$ represents movement of robot $p$ from $x$ to $y$; if there is a robot on point $y$, it must be removed before movement of robot $p$ from point $x$ to point $y$.

## 4. Distance Between Elements

Definition: A *MAP* of the set $X$ with respect to point $x$ and element $p$ for the Complex System is the mapping [2]:

$$MAP_{x,p} : X \rightarrow Z_+ \quad (2)$$

(where $x \in X, p \in P$ and $Z_+$ is the set of all non-negative integer numbers ).

*MAP* is constructed as follows:

Consider a family of reachable areas from the point $x$, i.e. a finite set of the following nonempty subsets $\{M^k_{x,p}\}$ of $X$ (as show in figure 1):

$k = 1: M^k_{x,p}$ is a set of points $m$ reachable in one step from $x$ when $R_p(x,m)=Ture$;

$k > 1: M^k_{x,p}$ is a set of points reachable in $k$ steps and not reachable in $k-1$ steps, i.e. points $m$ are only reachable from points $M^{k-1}_{x,p}$.

Let:

$MAP_{x,p}(y) = k$, for $y$ from $M^k_{x,p}$, denotes the number of steps from $x$ to $y$ .

For other points we have

$MAP_{x,p}(y) = 2n$ if $y \neq x$ and

$$MAP_{x,p}(y) = 0 \quad \text{if } y = x \quad (3)$$

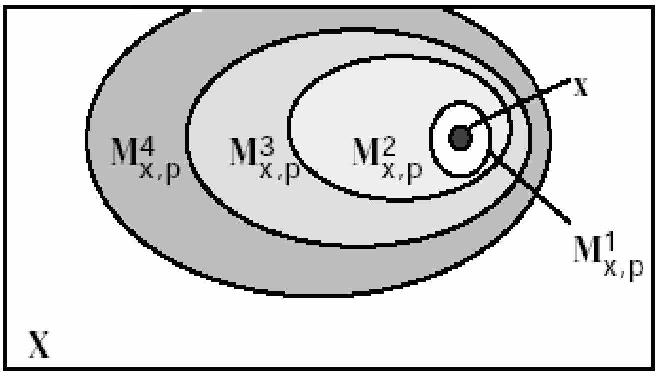


**Fig 1.** Interpretation of the family of reachable areas

## 5. Trajectories

Definition: A trajectory for an element $p \in P$ with the origin at $x \in X$ and the destination at $y \in X$ $(x \neq y)$ with a length $l$ is the following formal string of symbols $a(x)$ with points of $X$ as parameters:

$$t_0 = a(x)a(x_1)...a(x_l) \quad (4)$$

Where $x_l = y$ and each successive point $x_{i+1}$ is reachable from the previous point $x_i$, i.e. $R_p(x_i, x_{i+1})$ holds for $i = 0,1,...,l-1$; element $p$ stands at the point $x$: $ON(p) = X$ [2].

$t_p(x, y, l)$ is the set of all trajectories for element $p$, originated at $x$ with destination $y$ and length $l$.

Definition: the shortest trajectory $t$ of $t_p(x, y, l)$ is the trajectory of minimum length for the given origin $x$, destination $y$, and element $p$.

The above and following Properties of Complex System allow us to construct formal grammars for generating the shortest trajectories.

Definition: An admissible trajectory of degree $k$ is the trajectory $t_0$ which can be divided into $k$ shortest trajectories; more precisely there exists a subset $\{x_{i_1}, x_{i_2}, ..., x_{i_{k-1}}\}, i_1 < i_2 < ... < i_{k-1}, k \le l$ such that corresponding substrings $a(x_0)...a(x_{i_1}), a(x_{i_1})...a(x_{i_2}),..., a(x_{i_{k-1}})...a(x_l)$ are the shortest trajectories (as shown in figure 2).

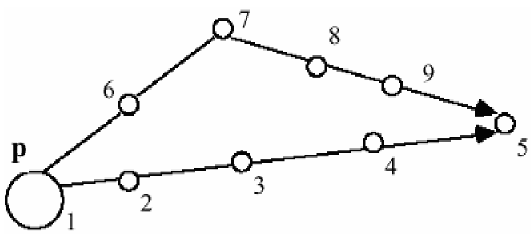


**Fig 2.** An interpretation of shortest and admissible trajectories

## 6. Generating Trajectories

Consider the following control grammar for the Complex System with symmetric relation of reachable $R_p$, as summarized in table 1 [2].

**Table 1.** summary of Grammar of shortest trajectories $\mathbf{G_t}$

| L | Q | Kernel, $\pi_k$ | $\pi_n$ | $F_T$ | $F_F$ |
|---|---|---|---|---|---|
| 1 | $Q_1$ | $S(x,y,l) \rightarrow A(x,y,l)$ | | two | $\phi$ |
| $2_i$ | $Q_2$ | $A(x,y,l) \rightarrow a(x)A(next_i(x,l),y,f(l))$ | | two | 3 |
| 3 | $Q_3$ | $A(x,y,l) \rightarrow a(y)$ | | $\phi$ | $\phi$ |

*Pred*=$\{\mathbf{Q_1}, \mathbf{Q_2}, \mathbf{Q_3}\}$

$\mathbf{Q_1}(x,y,l) = (MAP_{x,p}(y) = l) \quad (0<l<n)$

$\mathbf{Q_2}(l) = (l \ge 1)$

$\mathbf{Q_3} = T$

($n = |X|$, number of points in $X$ )

$f(l) = l\text{-}1 \qquad D(f) = Z_+ / \{0\}$

$\mathbf{L} = \{1,3\} \cup \text{two}$, two=$\{2_1, 2_2, ..., 2_n\}$

where **L** is a finite set called the set of labels;

$\mathbf{Q_i}$ represents the condition of applicability of productions (antecedents)

If $\mathbf{Q_i} = \boldsymbol{T}$ then $\boldsymbol{F_T}$ (a permitted subset of **L**) is reachable in the next step;

$\boldsymbol{F_F}$ is analogous to $\boldsymbol{F_T}$ in the case of $\mathbf{Q} = \boldsymbol{F}$.

Where $\boldsymbol{T}$ is true and $\boldsymbol{F}$ is false

**At the beginning of derivation:** $x = x_0$, $y = y_0$, $l = l_0$, $x_0 \in X$, $y_0 \in X$, $l_0 \in Z_+$, $p \in P$

***next_i*** is defined as follows:

$\text{SUM} = \{v | v \in X, \text{MAP}_{x_0,p}(v) + \text{MAP}_{y_0,p}(v) = l_0\}$

$\text{ST}_k(x) = \{v | v \in X, \text{MAP}_{x,p}(v) = k\}$

$\text{MOVE}_l(x) = \text{ST}_1(x) \cap \text{ST}_{l_0-l+1}(x_0) \cap \text{SUM}$

IF

$\text{MOVE}_l(x) = \{m_1, m_2, ..., m_r\} \neq \{\}$

THEN

$\text{next}_i(x, l) = m_i \quad$ for $i \le r$ and

$\text{next}_i(x, l) = m_r \quad$ for $r < i \le n$

OTHERWISE

$\text{next}_i(x, l) = x$

**Theorem:** A shortest path from point $x$ to point $y$ with the length $l_0$ for an element $p$ on $x$, $ON(p) = x$, exists if and only if the distance between these two points is $l_0$:

$$MAP_{x_0,p}(y_0) = l_0 \quad (5)$$

Where $l_0 < 2n$. If the relation $R_p$ is symmetric, i.e. for all $x \in X, y \in X$ and $p \in P$,

$R_p(x,y) = R_p(y,x)$, then all of the shortest trajectories $t_p(x_0, y_0, l_0)$ can be generated by the grammar $G_t$ [3].

In this grammar, $MOVE_l(x)$ the intersection of three sets *SUM*, $ST_1(x)$ and $ST_{l_0-l+1}(x_0)$. If the intersection of these sets is empty, the moving element will remain in position $x_0$. The following example reveals this problem.

**Example:**

A military robot A is at position (1, 1, 1) with its target TA at position (8, 2, 2) surrounded by static obstacles.
Figure 3 shows the situations of robot and the target in a $8 \times 8 \times 8$ cubic (grid). The robot can move to one of its 26 adjacent points.

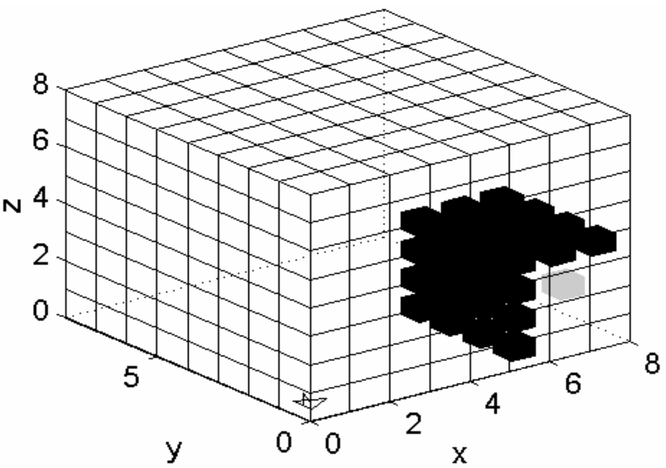

**Fig 3.** Initial state of robot and its target

$MAP_{(1,1,1),A}$ represents the reachable steps for robot A as shown in figures 4, 5.

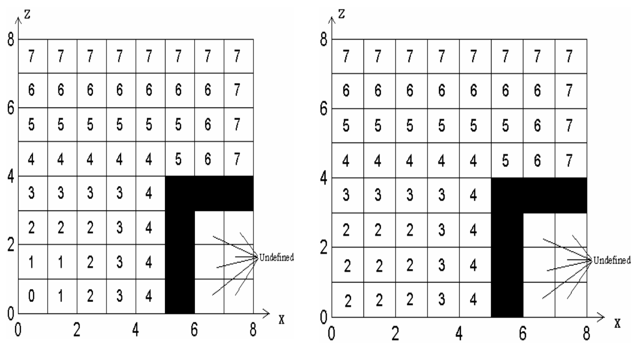


**Fig 4.** $MAP_{(1,1,1),A}$ (projection to XZ plane) left for Y=1 and right for Y=3

The value of $MAP_{(1,1,1),A}$ for Y=2 is similar to figure 4 (left) except in cell (X=1, Z=1) must be 1.

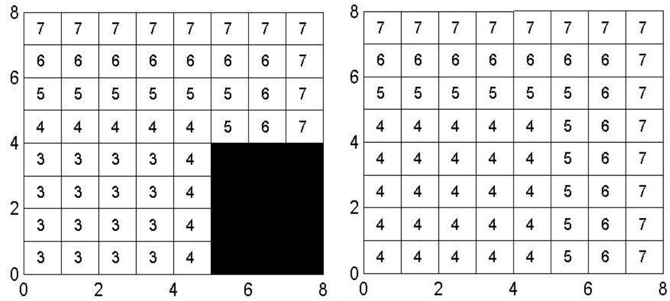

**Fig 5.** $MAP_{(1,1,1),A}$ (projection to XZ plane) left for Y=4 and right for Y=5

$MAP_{(1,1,1),A}$ for Y=6,7,8 is achieved in the same manner.

Note that $MAP_{(8,2,2),TA}$ is defined at points, where $MAP_{(1,1,1),A}$ is undefined.

Hence none of elements of *SUM* are defined. In this case the robot A cannot find any path to the target TA and it remains unmovable at point (1, 1, 1). Using the following method, the robot tries to destroy one of the surrounded obstacles from the weakest point for approaching the target.

## 7. Fuzzy Algorithm for Destroying One of the Surrounding Obstacles

In the first step, the surrounding obstacles are detected. Then *SUM*, $ST_1(x)$ and $ST_{l_0-l+1}(x_0)$ are calculated by the algorithm without considering the obstacles. In this way the robot starts to move towards the target until it reaches the surrounding obstacles. In this step, the fuzzy procedure will be used to detect the weakest point to be destroyed. Figure 6 shows the flowchart of this procedure.

Figures 7, 8 and 9 show the $MAP_{(1,1,1),A}$, $MAP_{(8,2,2),TA}$ and *SUM*, without taking into account the surrounding obstacles.

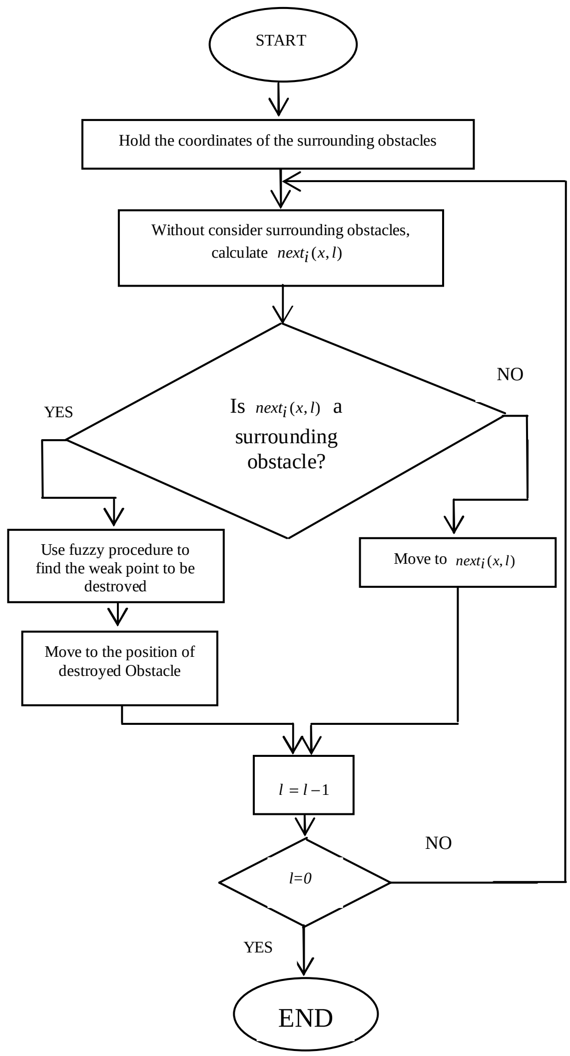


**Fig 6.** Flowchart of destroying procedure

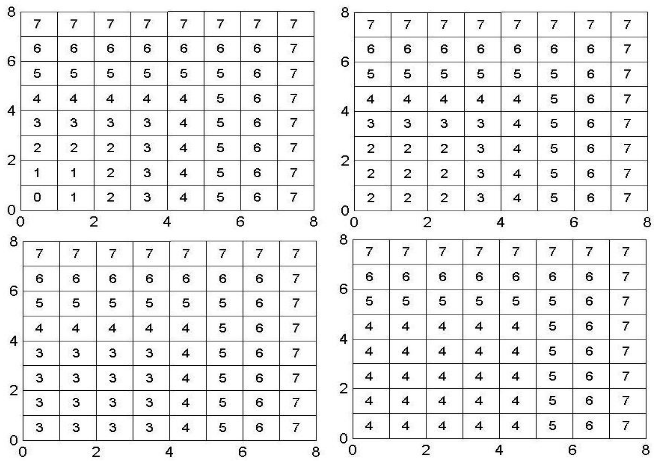


**Fig 7**. $MAP_{(1,1,1),A}$ without considering surrounding obstacles (projection to XZ plane)

Left_top:Y=1 Right_top:Y=3

Left_bottom:Y=4 Right_bottom:Y=5

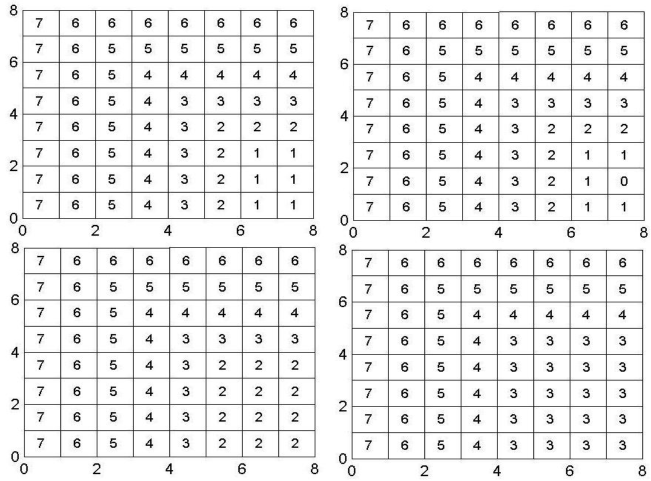


**Fig 8**. $MAP_{(8,2,2),\mathrm{TA}}$ without considering surrounding obstacles (projection to XZ plane)

Left_top:Y=1 Right_top:Y=2

Left_bottom:Y=4 Right_bottom:Y=5

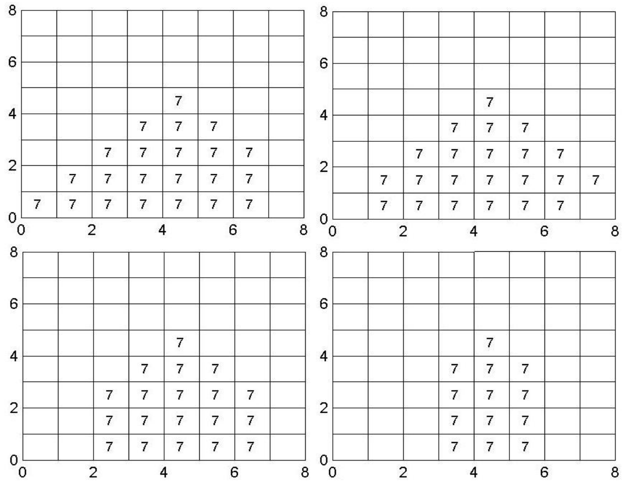


**Fig 9**. *SUM* without considering surrounding obstacles (projection to XZ plane)

Left_top:Y=1 Right_top:Y=2

Left_bottom:Y=3 Right_bottom:Y=4

As it is seen, 7 steps are needed to move from point A to point TA when the surrounding obstacles are not considered. As shown in figure 10 at point (1, 1, 1) $ST_{7-7+1}(1,1,1)=ST_1(1,1,1)$. The intersection of three sets $SUM$, $ST_{7-7+1}(1,1,1)$, $ST_1(1,1,1)$ is shown in figure 11.

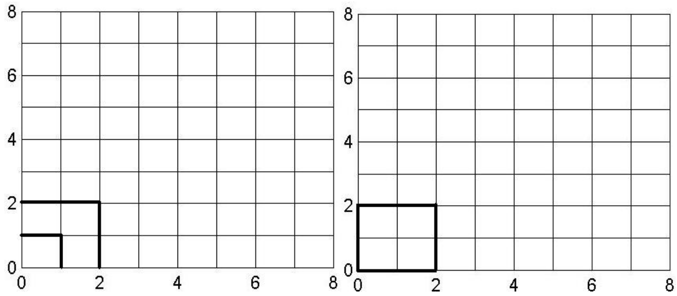

**Fig 10**. Set of $ST_1(1,1,1)$ (projection to XZ plane)
Left:Y=1 Right:Y=2

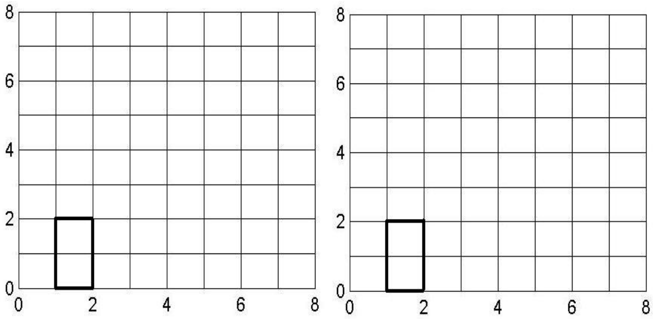

**Fig 11**. Intersection of three sets $ST_1(1,1,1)$, *SUM*, $ST_{7-7+1}(1,1,1)$ (projection to XZ plane)
Left:Y=1 Right:Y=2

Consequently,
$MOVE_7(1,1,1) = \{(2,1,1),(2,1,2),(2,2,1),(2,2,2)\}$

Therefore

$next((1,1,1),7) = (2,1,1)$ $\quad next((1,1,1),7) = (2,1,2)$

$next((1,1,1),7) = (2,2,1)$ $\quad next((1,1,1),7) = (2,2,2)$

Since the number of different values of *next* is equal to 4, at this step we can branch one of them, by applying productions $2_1$, $2_2$, $2_3$, $2_4$ simultaneously (see table 1).

Suppose point (2,1,1) is chosen. Now robot is in (2,1,1) state. $MOVE_6(2,1,1)$ is determined by using the grammar of table 1. The two sets of $ST_1(2,1,1)$ and $ST_{7-6+1}(1,1,1) = ST_2(1,1,1)$ are shown in figures 12, 13.

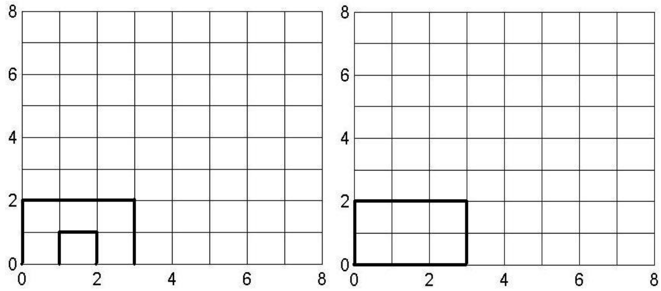

**Fig 12**. Set $ST_1(2,1,1)$ (projection to XZ plane)
Left:Y=1 Right:Y=2

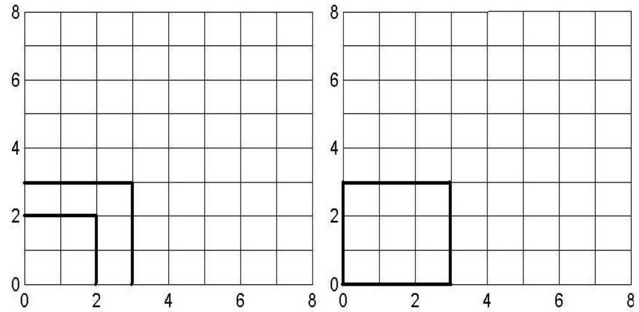

**Fig 13**. Set $ST_2(1,1,1)$ (projection to XZ plane)
Left:Y=1, Y=2 Right:Y=3

The set *SUM* is fixed and is shown in figure 9. Hence, $MOVE_6(2,1,1)$ is the intersection of the sets shown in figures 9,12 and 13. Then

$next((2,1,1),6) = (3,1,1)$ $\quad next((2,1,1),6) = (3,1,2)$
$next((2,1,1),6) = (3,2,1)$ $\quad next((2,1,1),6) = (3,2,2)$

Suppose point (3,2,2) is chosen. And robot moves to this point. Continuing the algorithm, a set of paths from A to surrounding obstacle will be obtained, where one of them

$t_A$= a(1, 1 ,1) a(2 , 1, 1) a(3, 2, 2) a(4, 2 ,2) a(5, 2, 2)
is shown in figure 14.

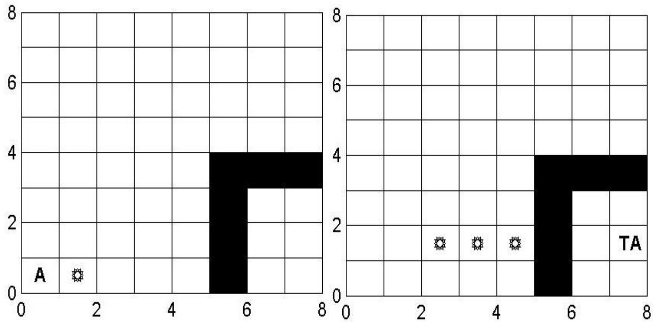


**Fig 14**. Set of paths from A to TA, without taking account the surrounding obstacles

In this step the fuzzy rule base procedure is used to choose one of the nine points (6, 1, 1), (6, 2, 1), (6, 3, 1), (6, 1, 2), (6, 2, 2), (6, 3, 2), (6, 1, 3), (6, 2, 3), (6, 3, 3) towards target TA.

## 8. Fuzzy Rule Base

Fuzzy decision-making is a powerfull procedure for decision-making in imprecise environments [8]. In this work the destroying ability of the robot is related to two main criteria

a) Self protection of obstacles.
b) Robustness of the obstacle.

Table 2 represents the rule bases for these relations, where

***VV*** ≡ Veryvery      ***V*** ≡ Very
***S*** ≡ slightly      ***EXT*** ≡ extremely

**Table 2.** Fuzzy rule base

| Self Protect / Robustness | LOWEST | LOW | NLOWN HIGH | HIGH | HIGHEST |
|---|---|---|---|---|---|
| LOWEST | ***VV*** (***EXT***_easy) | ***S*** (***EXT***_easy) | easy | ***S*** (diff) | ***VV*** (diff) |
| LOW | ***V*** (***EXT***_easy) | ***VV*** (easy) | ***S*** (easy) | diff | ***S*** (***EXT_diff***) |
| NLOWN HIGH | (***EXT***_easy) | ***V*** (easy) | ndif_ neasy | ***V*** (diff) | (***EXT_diff***) |
| HIGH | ***S*** (***EXT***_easy) | easy | ***S*** (diff) | ***VV*** (diff) | ***V*** (***EXT_diff***) |
| HIGHEST | ***VV*** (easy) | ***S*** (easy) | diff | ***S*** (***EXT_diff***) | ***VV*** (***EXT_diff***) |

The membership functions for fuzzy values of table 2 are shown in figure 15.

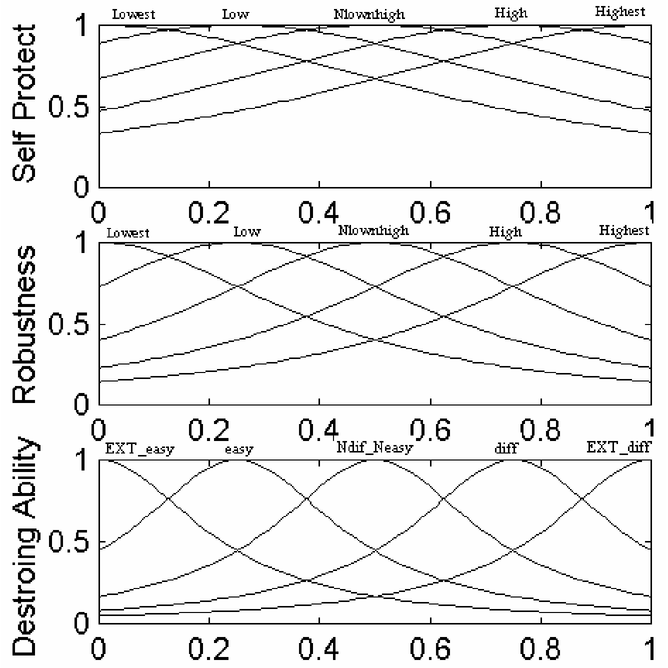


**Fig 15.** Membership functions for fuzzy values

Suppose the current self protection and robustness for the mentioned nine points be as presented in table 3.

**Table 3.** Fuzzy values for self protection and robustness of nine points

| $next_i(x,l)$ | (6, 1, 1) | (6, 2, 1) | (6, 3, 1) |
|---|---|---|---|
| Self Protect | HIGHEST | HIGH | HIGHEST |
| Robustness | HIGH | LOW | LOW |
| $next_i(x,l)$ | (6, 1, 2) | (6, 2, 2) | (6, 3, 2) |
| Self Protect | HIGHEST | NLOWNHIGH | NLOWNHIGH |
| Robustness | LOW | HIGHEST | HIGHEST |
| $next_i(x,l)$ | (6, 1, 3) | (6, 2, 3) | (6, 3, 3) |
| Self Protect | LOW | NLOWNHIGH | HIGH |
| Robustness | LOW | LOW | LOW |

The defuzzified decision values for these points are shown in table 4.

**Table 4.** Defuzzified decision values for nine points

| *next*((5,2,2),3) | (6, 1, 1) | (6, 2, 1) | (6,3,1) |
|---|---|---|---|
| Decision value | 0.843 | 0.25 | 0.301 |
| *next*((5,2,2),3) | (6, 1, 2) | (6, 2, 2) | (6,3,2) |
| Decision value | 0.301 | 0.865 | 0.865 |
| *next*((5,2,2),3) | (6, 1, 3) | (6, 2, 3) | (6,3,3) |
| Decision value | 0.183 | 0.201 | 0.25 |

Therefore point (6, 1, 3) is destroyed and the robot moves through this point toward it's target.
The final path from A to TA will be

$t_A$= a(1, 1 ,1) a(2 , 1, 1) a(3, 2, 2) a(4, 2 ,2) a(5, 2, 2) a(6, 1, 3) a(7, 1,2) a(8, 2, 2)

## 9. Conclusion

Linguistic Geometry provides a hierarchy of formal languages for logical reasoning on multiagent systems. This method is applied for solving computationally hard search problems.
In this paper the fuzzy-Linguistic Geometry is used for 3D path planning of a moving robot when the robot or the target is surrounded by obstacles. The mutual advantages of Linguistic Geometry for decreasing the calculation time, and fuzzy theory for increasing the capability of the procedure for applications in real world, are benefited.
Two main criteria, Self protect and Robustness, are used for decision-making of moving robot to choose the appropriate point for penetrating the surrounding obstacles. The type and number of

criteria depends on the type of problem in real applications.
This method may be used in industrial large scale and military systems.

## References


[1] B. Stilman, “Managing Search Complexity in Linguistic Geometry”, IEEE Trans. on Systems, Man, and Cybernetics Part B: Cybernetics, Vol. 27, No. 6, pp. 978-998, Dec. 1997

[2] B. Stilman, “Linguistic Geometry Tools: Selected Topics”,Proc. of the IEEE Int Conf. on Systems, Man, and Cybernetics, La Jolla , Ca , Oct . 1998

[3] B. Stilman, “A Linguistic Approach To Geometric Reasoning ”, An international Journal: Computers & Mathematics with Applications, Vol.26, No.7, pp.29-58, 1993

[4] B. Stilman, “Network Languages for Concurrent Multiagent Systems”, An International Journal Computers & Mathematics with Applications, Vol. 34, No. 1, pp. 103 – 136, 1997

[5] E. Skhisov, B. Stilman, “No Search Approach in Linguistic Geometry: First Implementation”, Proc. of the IEEE Int. Conf. on Systems, Man, and Cybernetics, La Jolla, Ca, Oct. 1998

[6] B. Stilman, “A Linguistic Geometry for Space Applications”, Proc.of the 1994 Goddard Conf. On Space Application of Artificial Intelligence, pp.87 –101, NASA Goddard Space Flight Center , Greenbelt ,MD, USA, May 1994

[7] B. Stilman, “ Heuristic Networks for Space Exploration, Telematics and Informatics”, An Int. Journal on Telecommunication & Information Technology, Vol. 11, NO. 4, pp. 403- 428, 1994

[8] Lee, J., Jong-Yih Kuo; Huang, W.T.,“Fuzzy Decision Making Through Relationships Analysis Between Criteria”, IEEE Proceedings of the 1996 Asian pp. 296 –301